I laboration 1 gjorde ni en funktion `polylen` som innehöll programkoden för beräkning av längden av ett polygontåg

```
n=length(x);
L=0;
for i=1:n-1
    L=L+sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2);
end
```

Har vi bildat koordinater för ett polygontåg i två vektorer `x` och `y` så kan vi beräkna längden av polygontåget enligt

```
>> x=[1 3 3 1]; y=[1 1 2 1];
>> L=polylen(x,y)
```

Längden beräknas utan att vi behöver skriva ned koden som gör beräkningen.

Skall vi beräkna längden av ett nytt polygontåg så bildar vi bara två nya vektorer med koordinaterna och använder `polylen` igen. Vi har gjort en funktion som fungerar som andra färdiga funktioner i MATLAB.

Vi kallade vektorerna för `x` och `y` men vi kunde lika gärna använt andra variabelnamn. T.ex.

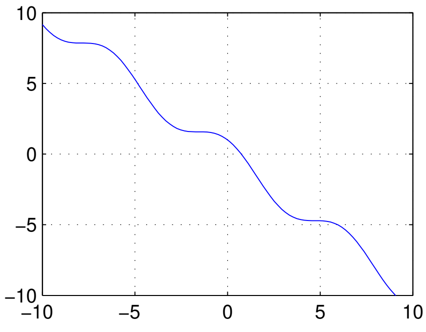
```
>> u=[1 3 3 1]; v=[1 1 2 1];
>> s=polylen(u,v)
```

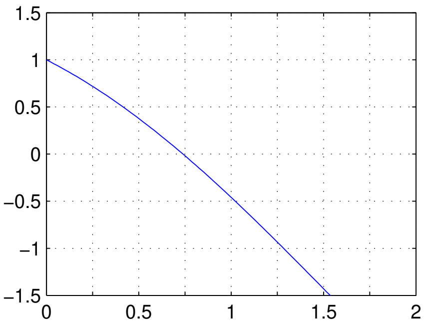
skulle fungera lika bra (vi bytte även variabelnamn på längden). Funktionen `polylen` fungerar precis som förut.

I laboration 2 skall vi göra en funktion `min_newton` som skall innehålla programkoden för Newtons metod. Sedan kan vi använda `min_newton` för att beräkna nollställen till olika funktioner $f(x)$.

Som exempel beräknar vi nollställena till funktionen $f(x) = \cos(x) - x$. Vi börjar med att rita en graf av funktionen så att vi ser hur många nollställen vi har och ungefär var de ligger genom att i MATLAB beskriva funktionen $f(x)$ (och dess derivata $f'(x)$) och rita grafen med

```
>> f=@(x)cos(x)-x; Df=@(x)-sin(x)-1;
>> x=linspace(-10,10);
>> plot(x,f(x)), grid on
```





Vi ser grafen ovan till vänster.

Som vi ser finns det ett nollställe och vi ritar om grafen med ett mindre intervall för att bättre se var nollstället ligger.

```
>> x=linspace(0,2);
>> plot(x,f(x))
```

Detta ger grafen ovan till höger. Vi ser att vi har ett nollställe nära $x_0 = 0.75$. När ni väl har gjort funktionen `min_newton` skall vi kunna beräkna nollstället noggrant enligt

```
>> x0=0.75;
>> w=min_newton(f,Df,x0,0.5e-8)
w=
   0.7391
```

Skall vi sedan beräkna nollställen till en ny funktion så är det bara att beskriva nya funktionen och dess derivata, rita graf och läsa av approximation av nollställe och använda `min_newton` igen.

Den programkod vi skall "bädda in" eller "packa in" i `min_newton` är den som beskriver Newtons metod, dvs.

```
kmax=10;
for k=1:kmax
    h=-f(x)/Df(x);
    x=x+h;
    if abs(h)<tol, break, end
end
```

För att göra det hela lite lättare finns ett programskal på kurshemsidan att utgå ifrån.

Ni kommer också göra en modifierad version av `min_newton` som vi kallar `min_modnewton` där derivatan approximeras i funktionen. Vi kommer at ha gjort en slags egen version av `fzero`.

Avslutningsvis skall vi göra funktionen `min_integral` för att beräkna integraler. Detta blir då en egen verison av `integral`.

Avsikten med laborationen är dels att vi skall förstå ungefär hur färdiga program är uppbyggda, dels ge lite träning i att programmera.